CLAIMS

I claim:

41. A collection command applicator process for applying computer commands to one or more collections, to be performed on or with the aid of a computer configured with collection command applicator software means for applying computer commands to collections, comprising the following steps:

receiving a list of one or more collections from a request originator;

applying one or more computer commands to collections within said list of collections; and

returning results from said applying computer commands to said request originator,

wherein collections are data structures comprised of a collection specifier and collection content containing zero or more collection content files, and wherein a collection specifier contains information about a collection instance,

thereby solving the general collection command applicator problem.

42. The process of claim 41, further comprising the step of:

sorting said list of collections into a visit order before applying computer commands to collections within said list of collections,

thereby ensuring that commands are applied in proper visit order according to processing interdependencies that may exist among processed collections, and thereby providing a solution to the collection visit order problem.

43. The process of claim 42, wherein

said step of sorting said list of collections uses information selected from a group consisting of collection specifier information and collection type definition information and collection content information,

thereby enabling human knowledge workers to correctly process large sets of collections that require dynamically calculated, complex execution visit orderings that must be based on detailed collection information.

44. The process of claim 41, wherein

(a) said step of applying one or more computer commands uses a command execute parallel means to apply commands,

thereby solving the parallel collection command execution problem.

45. The process of claim 41, wherein

(a) said step of applying one or more computer commands uses an indirect command execution means selected from a group consisting of command execution sequential indirect means and command execution parallel indirect means,

thereby creating an efficient, reusable, and persistent way of applying arbitrary commands to a set of collections without incurring collection list production costs for each future command application to the same set of collections.

46. The process of claim 41, wherein

(a) said step of applying one or more computer commands uses nearby execution directory techniques to calculate ultimate command execution directories,

thereby solving the nearby execution directory problem.

47. A programmable collection command applicator device for applying computer commands to one or more collections configured with collection command applicator software means for applying computer commands to collections, whose actions are directed by software executing a process comprising the following steps:

receiving a list of one or more collections from a request originator;

applying one or more computer commands to collections within said list of collections; and

returning results from said applying computer commands to said request originator,

wherein collections are data structures comprised of a collection specifier and collection content containing zero or more collection content files, and wherein a collection specifier contains information about a collection instance,

thereby solving the general collection command applicator problem.

48. The programmable device of claim 47, further comprising the step of:

sorting said list of collections into a visit order before applying computer commands to collections within said list of collections,

thereby solving the collection visit order problem.

49. The programmable device of claim 48 wherein

said step of sorting said list of collections uses information selected from a group consisting of collection specifier information and collection type definition information and collection content information,

thereby enabling human knowledge workers to correctly process large sets of collections that require dynamically calculated, complex execution visit orderings that must be based on detailed collection information.

50. The programmable device of claim 47, wherein

said step of applying one or more computer commands uses a command execute parallel means to apply commands,

thereby using parallel processing techniques to apply commands to collections in a minimum amount of time, and thereby providing a solution to the parallel collection command execution problem.

51. The programmable device of claim 47, wherein

said step of applying one or more computer commands uses an indirect command execution means selected from a group of command execution sequential indirect means and command execution parallel indirect means,

thereby creating an efficient, reusable, and persistent way of applying arbitrary commands to a set of collections without incurring collection list production costs for each future command application to the same set of collections.

52. The programmable device of claim 47, wherein

said step of applying one or more computer commands uses nearby execution directory techniques to calculate ultimate command execution directories,

thereby solving the nearby execution directory problem.

53. A computer readable memory, encoded with data representing a collection command application computer software means that can be used to direct a computer when used by the computer, comprising:

means for receiving a list of one or more collections from a request originator;

means for applying one or more computer commands to collections within said list of provided collections; and

means for returning results from said applying computer commands to said request originator,

wherein collections are data structures comprised of a collection specifier and collection content containing zero or more collection content files, and wherein a collection specifier contains information about a collection instance,

thereby solving the general collection command applicator problem.

54. The computer readable memory of claim 53, further comprising:

means for sorting said list of collections into a visit order before applying computer commands to collections within said list of collections,

thereby providing a solution to the collection visit order problem.

55. The computer readable memory of claim 54, wherein

said means for sorting said list of collections uses information selected from a group of collection specifier information and collection type definition information and collection content information,

thereby enabling human knowledge workers to correctly process large sets of collections that require dynamically calculated, complex execution visit orderings that must be based on detailed collection information.

56. The computer readable memory of claim 53, wherein

(a) said means for applying one or more computer commands uses a command execute parallel means to apply commands,

thereby solving the parallel collection command execution problem.

57. The computer readable memory of claim 53, wherein

(a) said means for applying one or more computer commands uses nearby execution directory techniques to calculate ultimate command execution directories,

thereby solving the nearby execution directory problem.